AMENDMENTS TO THE SPECIFICATION

Please amend the paragraph that starts on page 9, line 30, and ends on page 10, line 12, as follows:

SAN **100** is a high-bandwidth, low-latency network interconnecting nodes within the distributed computer system. A node is any component attached to one or more links of a network and forming the origin and/or destination of messages within the network. In the depicted example, SAN **100** includes nodes in the form of host processor node **102**, host processor node **104**, redundant array independent disk (RAID) subsystem node **106**, and I/O chassis node **108**. The nodes illustrated in **Figure 1** are for illustrative purposes only, as SAN **100** can connect any number and any type of independent processor nodes, such as node 110, I/O adapter nodes, and I/O device nodes. Any one of the nodes can function as an endnode, which is herein defined to be a device that originates or finally consumes messages or frames in SAN **100**.

Please amend the paragraphs that start on page 27, line 28, and end on page 28, line 27, as follows:

In **Figure 8**, a portion **800** of a distributed computer system is depicted to illustrate an example request and acknowledgment transaction. The distributed computer system in **Figure 8** includes a host processor node **802** and a host processor node **804**. Host processor node **802** includes a host channel adapter **806**. Host processor node **804** includes a host channel adapter **808**. The distributed computer system in **Figure 8** includes a SAN fabric **810**, which includes a switch **812** and a switch **814**. The SAN fabric includes a link coupling host channel adapter **806** to switch **812**; a link coupling switch **812** to switch **814**; and a link coupling host channel adapter **808** to switch **814**.

In the example transactions, host processor node **802** includes a client process A **816**. Host processor node **804** includes a client process B **818**. Client process A **816** interacts with host channel adapter hardware **806** through queue pair **820** [[824]]. Client process B **818** interacts with hardware channel adapter hardware **808** through queue pair

822 [[828]]. Queue pairs **820** [[824]] and **822** [[828]] are data structures that include a send work queue **824, 828** and a receive work queue **826, 830**.

Process A **816** initiates a message request by posting work queue elements to the send queue **824** of queue pair **820** [[824]]. Such a work queue element is illustrated in **Figure 4**. The message request of client process A **816** is referenced by a gather list contained in the send work queue element. Each data segment in the gather list points to a virtually contiguous local memory region, which contains a part of the message, such as indicated by data segments 1, 2, and 3, which respectively hold message parts 1, 2, and 3, in **Figure 4**.

Please amend the paragraph that starts on page 34, line 28, and ends on page 35, line 8, as follows:

Consumers **1103** and **1105** represent applications or processes that employ the other layers for communicating between endnodes. Transport layer **1104** provides end-to-end message movement. In one embodiment, the transport layer provides four types of transport services as described above which are reliable connection service; reliable datagram service; unreliable datagram service; and raw datagram service. Network layer **1106** performs packet routing through a subnet or multiple subnets to destination endnodes. Link layer **1108** performs flow-controlled **1120**, error checked, and prioritized packet delivery across links.